AMENDMENTS TO THE CLAIMS

Please amend the claims as indicated in the following listing of all claims:

1. - 20. Canceled

B2

21. (New) A method comprising:

locking a resource to be accessed by an executable computer instruction, wherein locking the resource includes locking the resource prior to completing a determination of hazards related to the access.

22. (New) The method of claim 21 wherein locking the resource includes locking the resource prior to the executable computer instruction entering a trap stage of an instruction pipeline.

23. (New) The method of claim 21 wherein the executable computer instruction is an atomic instruction including a portion to lock the resource and a portion to unlock the resource.

24. (New) The method of claim 21 wherein determination of a hazard includes determining whether a read-after-write hazard exists.

25. (New) The method of claim 21 wherein locking the resource includes:

locking the resource during an effective address calculation stage of an instruction pipeline.

26. (New) The method of claim 21 wherein locking a resource includes locking at least a portion of a cache.

27. (New) The method of claim 21 wherein locking a resource includes locking at least one memory address.

28. (New) The method of claim 21 further comprising unlocking the resource no later than a time at which the executable computer instruction exits an instruction pipeline, regardless of whether the executable computer instruction is cancelled.

29. (New) The method of claim 28 wherein unlocking the resource includes:
unlocking the resource in the normal course of executing the computer instruction.

30. (New) The method of claim 28 wherein unlocking the resource includes:
preventing a write portion of the executable computer instruction from altering information held in at least a portion of the resource.

31. (New) The method of claim 30 wherein preventing a write portion from altering information includes suppressing writing a value to an architectural storage location.

32. (New) A processor comprising:
at least one processing core to lock a resource in response to an access by an executable computer instruction prior to completing a hazard determination related to the access.

33. (New) The processor of claim 32 further comprising a plurality of processing cores, wherein respective processing cores are adapted to lock the resource in response to access by respective executable computer instructions prior to completing respective hazard determinations.

34. (New) The processor of claim 32 wherein said at least one processing core is adapted to lock the resource prior to the executable computer instruction entering a trap stage of a pipeline.

35. (New) The processor of claim 32 wherein said at least one processing core is adapted to implement an atomic instruction, wherein implementing the atomic instruction includes locking the resource and unlocking the resource.

36. (New) The processor of claim 32 wherein said processing core locks the resource before it is determined if a read-after-write hazard exists.

37. (New) The processor of claim 32 wherein said processing core locks the resource during an effective address calculation stage of a pipeline.

38. (New) The processor of claim 32 further including a cache, and wherein locking a resource includes locking at least a portion of the cache.

39. (New) The processor of claim 32 wherein said processing core further includes an output coupled to a memory, and wherein locking a resource includes locking at least one memory address.

40. (New) The processor of claim 32 further comprising logic to unlock the resource no later than a time at which the executable computer instruction exits an instruction pipeline, regardless of whether the executable computer instruction is cancelled.

41. (New) The processor of claim 40 wherein the processor includes logic to prevent a write portion of the executable computer instruction from altering information held in at least a portion of the resource if the computer instruction is cancelled.

42. (New) A processor adapted to speculatively dispatch a load operation to a cache unit prior to determining whether read-after-write hazards associated with the load operation are present.

43. (New) The processor of claim 42 wherein the processor is adapted to lock a resource associated with the load operation concurrently with dispatching the load operation.

44. (New) The processor of claim 43 wherein the processor is further adapted to unlock the resource associated with the load operation no later than a time at which an instruction implementing the load operation exits an instruction pipeline, regardless of whether the instruction is cancelled before exiting the instruction pipeline.

45. (New) A processor comprising:

means for determining a hazard related to accessing a resource; and

means for locking a resource to be accessed by an executable computer instruction, wherein locking the resource includes locking the resource prior to completing a determination of hazards related to the access.

B2

46. (New) The processor of claim 45 wherein means for locking the resource includes means for locking the resource prior to the executable computer instruction entering a trap stage of an instruction pipeline.

47. (New) The processor of claim 45 wherein the executable computer instruction is an atomic instruction including a portion to lock the resource and a portion to unlock the resource.

48. (New) The processor of claim 45 wherein the means for determining a hazard includes means for determining whether a read-after-write hazard exists.

49. (New) The processor of claim 45 wherein the means for locking the resource includes:

means for locking the resource during an effective address calculation stage of an instruction pipeline.

50. (New) The processor of claim 45 wherein means for locking a resource includes means for locking at least a portion of a cache.

51. (New) The processor of claim 45 wherein means for locking a resource includes means for locking at least one memory address.

52. (New) The processor of claim 45 further comprising means for unlocking the resource no later than a time at which the executable computer instruction exits an instruction pipeline, regardless of whether the executable computer instruction is cancelled.

53. (New) The processor of claim 52 wherein means for unlocking the resource includes:

means for unlocking the resource in the normal course of executing the computer instruction.

B2

54. (New) The processor of claim 52 wherein means for unlocking the resource includes:

means for preventing a write portion of the executable computer instruction from altering information held in at least a portion of the resource.

55. (New) The processor of claim 54 wherein means for preventing a write portion from altering information includes means for suppressing writing of a value to an architectural storage location.